**A PROJECT REPORT**

**ON**

**Seq2Seq network with Attention for**

**Machine Translation**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (Rajasthan)**

*May 2023*

For the partial fulfillment of the course-

**Natural Language Processing- CS F429**

*Jatin Bansal 2020A4PS0241P*

*Prem Shankar 2020A4PS0557P*

Dataset:

The dataset used for this assignment was spanish to english dataset which is available in tensorflow library. The source language chosen in this assignment is Spanish and the target language is english.

Preprocessing:

The preprocessing involved removing any alphanumeric words from the dataset as it would have made the learning for the model more difficult.We have also removed punctuations like . and , . Also we have converted all the words in the dataset to lowercase and also expanded the contractions like the words can’t have been converted to cannot this is done so that the model gets better context.For better translation and for letting model understand the start and end of sentences we have added two unique tokens <sos> and <eos> which denote the start and end of sentences.

1. Tokenization:Once the preprocessing has been done we now move to tokenization which is assigning words to numbers since the model does not understand words it only understands numbers. In this assignment we have used word-level tokenization which means that each word has been assigned a number.Further down the model that tokens are passed through the embedding layer which gives embedding weights to each token.We have used pretrained embedding weights for this purpose and taken those from (“GoogleNews-vectors-negative300 ( word2vec )”) for getting the english embedding and for spanish (“dccuchile/spanish-word-embeddings: Spanish word embeddings computed with different methods and from different corpora”).For both the case the embedding dimension is kept as 300.
2. Stop words Removal:

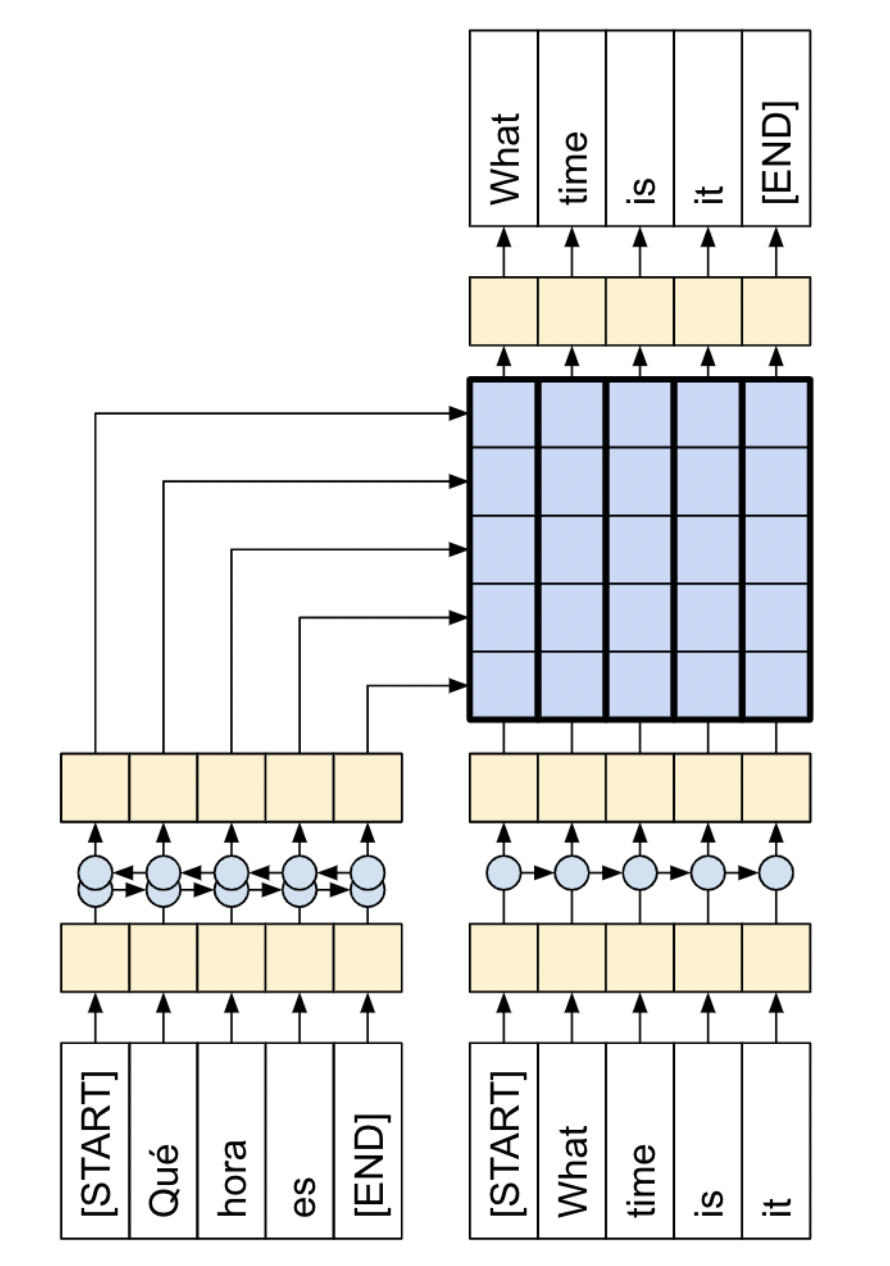
We have not removed stopwords from the dataset as it is required.

1. Padding:

Padding is done so that at each timestep the model receives a sequence of the same size and shape . To calculate the padding dimension we have calculated the maximum length of the English and spanish sequence present in the dataset and padded the sentences to max\_len+1.

We have padded the text with zeroes and we have used post-padding.

Seq2Seq model



This is the high-level overview of the Seq2Seq model we implemented in this assignment. Originally Seq2Seq model was proposed in paper published in 2014 named “*Sequence to Sequence Learning with Neural Networks*”. In this paper, the authors proposed the Deep Neural Networks (DNNs) can be used for their capabilities of arbitary parallel computations. In Seq2Seq model, the Encoder has layers of Bidirectional LSTM and decoder has layers of single directional LSTMs.

In this Assignment, we incorporated attention network to improve the efficiency of the traditional Seq2Seq network. In traditional Seq2Seq network,there are some limitations regarding the transfer of information from encoder to decoder and that is called information bottleneck.

With attention mechanism, we take output of each token of encoder, and passed it through the attention layer and then feed it to the decoder for the decoding the target language word.

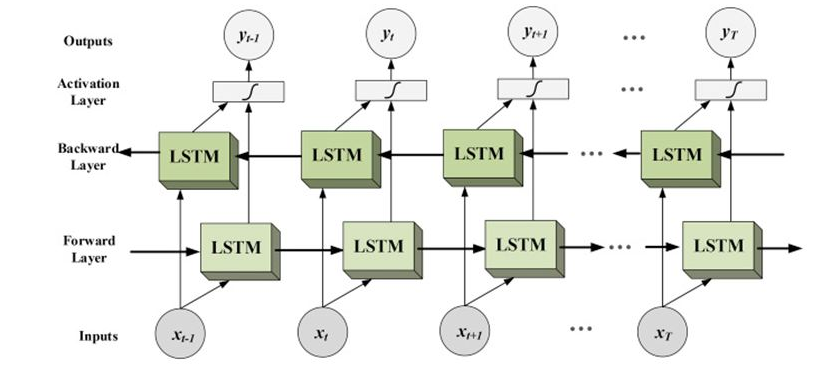
### 

### 

### **The encoder**

The goal of the encoder is to process the context sequence into a sequence of vectors that are useful for the decoder as it attempts to predict the next output for each timestep. Since the context sequence is constant, there is no restriction on how information can flow in the encoder, so use a bidirectional-LSTM to do the processing:



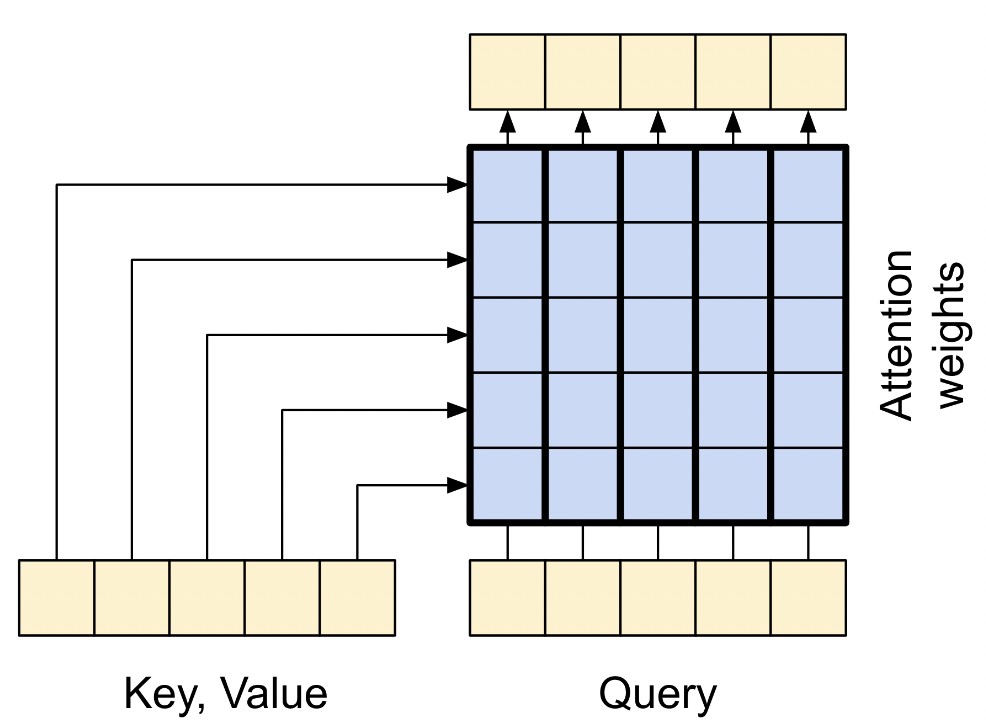


A Bidirectional LSTM

### **The attention layer**

The attention layer lets the decoder access the information extracted by the encoder. It computes a vector from the entire context sequence, and adds that to the decoder's output.

The simplest way you could calculate a single vector from the entire sequence would be to take the average across the sequence . An attention layer is similar, but calculates a **weighted** average across the context sequence. Where the weights are calculated from the combination of context and "query" vectors



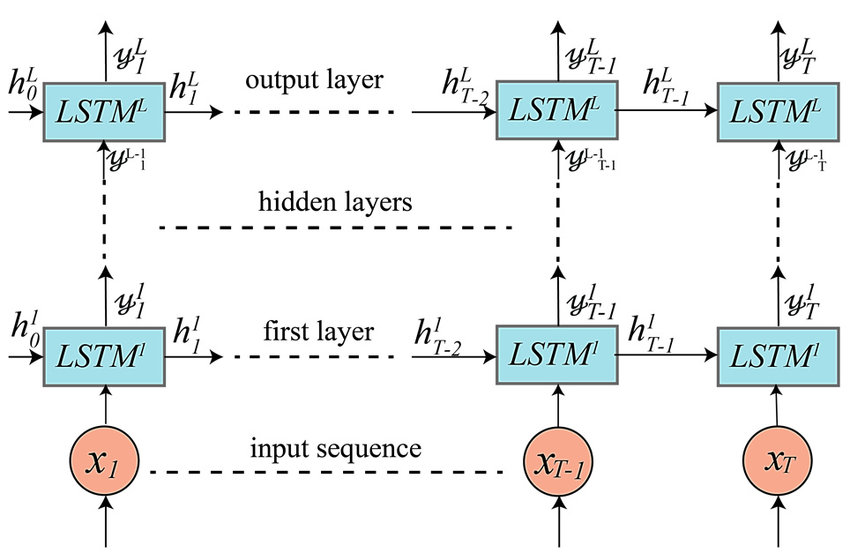
**The Decoder**

The decoder's job is to generate predictions for the next token at each location in the target sequence.

1. It looks up embeddings for each token in the target sequence.
2. It uses an LSTM to process the target sequence, and keep track of what it has generated so far.
3. It uses LSTM output as the "query" to the attention layer, when attending to the encoder's output.
4. At each location in the output it predicts the next token.

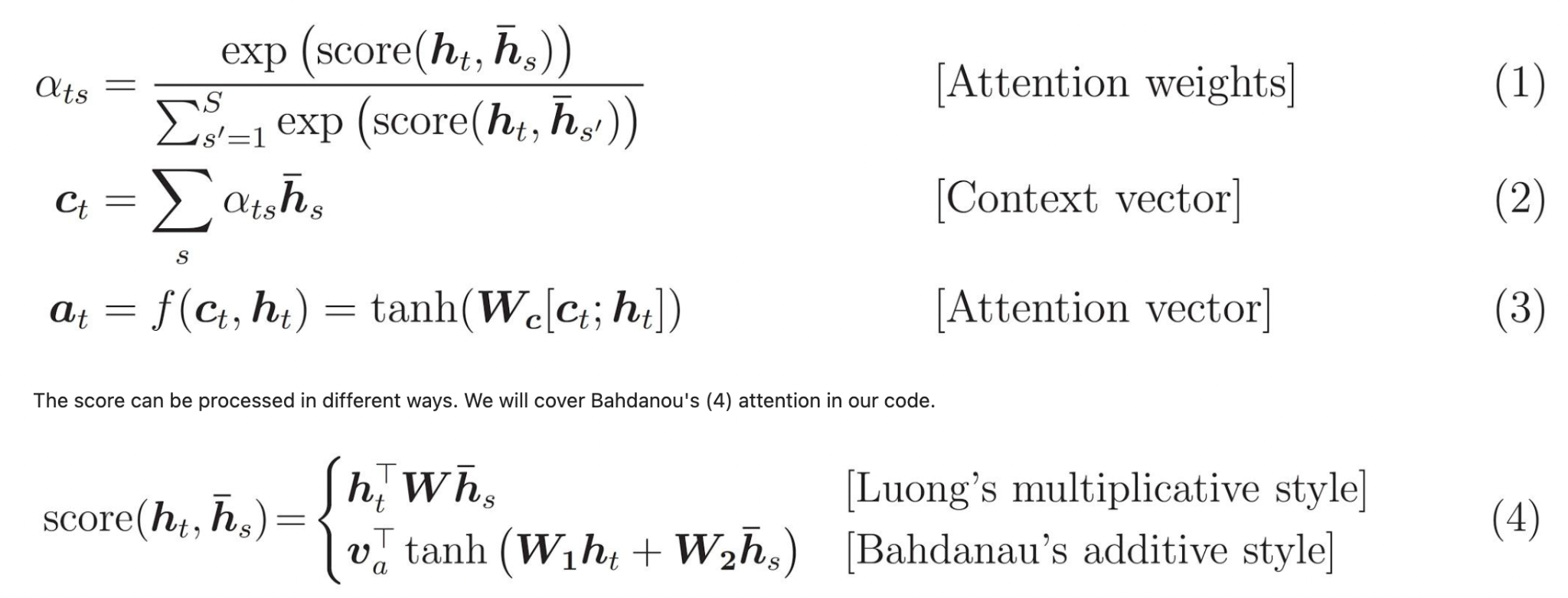
When training, the model predicts the next word at each location. So it's important that the information only flows in one direction through the model. The decoder uses a unidirectional (not bidirectional) LSTM to process the target sequence.

When running inference with this model it produces one word at a time, and those are fed back into the model.

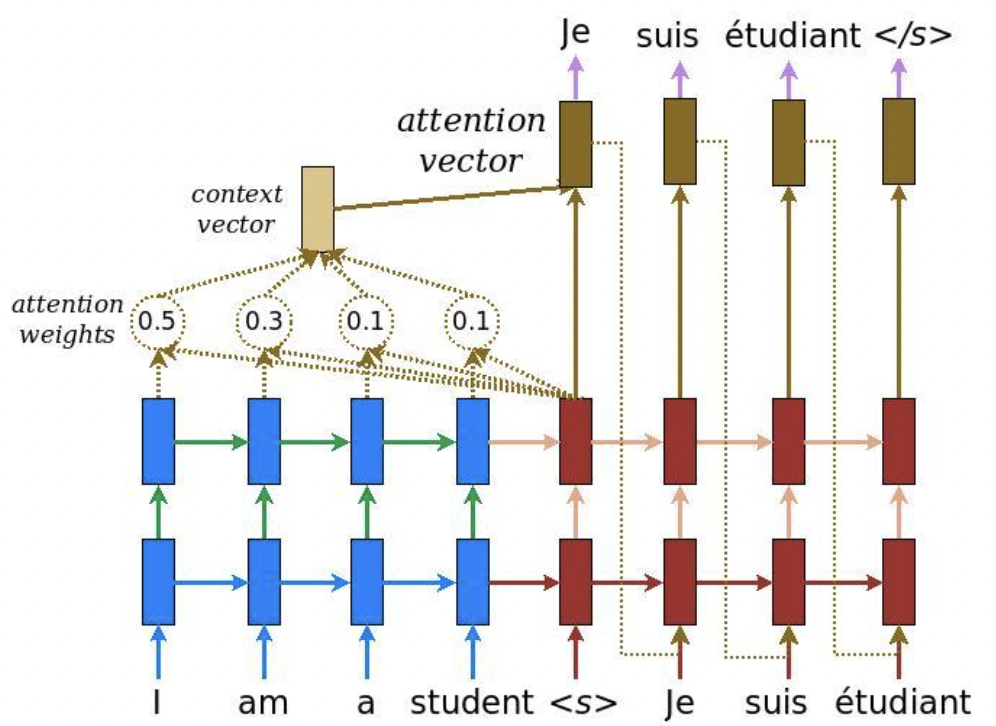


Unidirectional LSTM

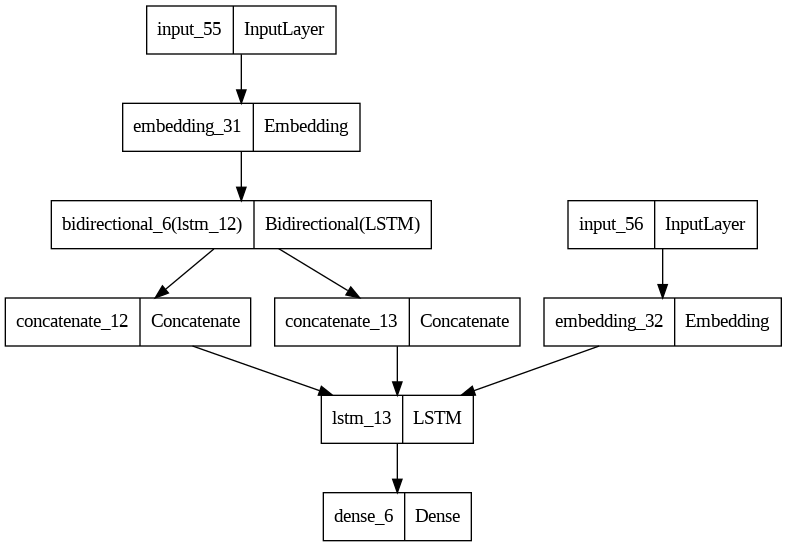
Equations involve in decoder with attention:



Overview of Seq2Seq network with attention:



**Our Model**

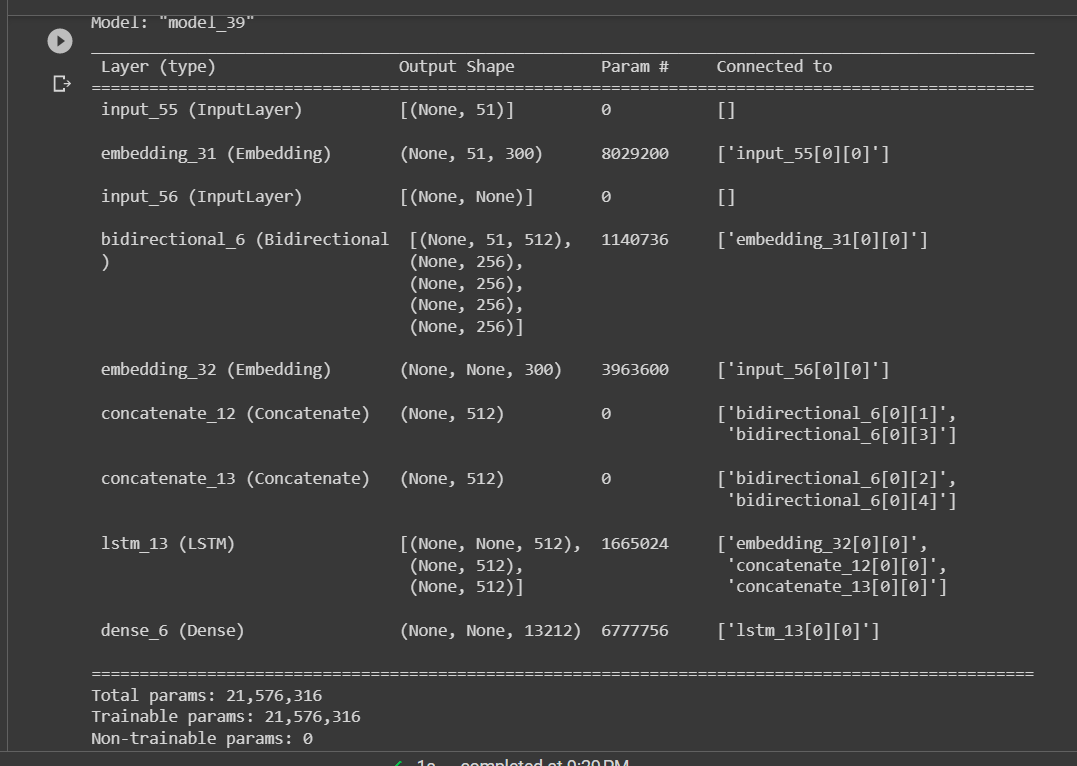
****

Above is the plot of the model which we have used in our assignment . It uses bi-directional lstm encoder and vanilla lstm decoder with additive attention .When the model was trained for 1 epoch due to computational constraints , the validation accuracy reached 0.90.

**Working of the model**

The source language in padded form is sent as input to the bidirectional lstm layer after it passes through the embedding layer which turns the tokens in the sequence to a vector of 300-dimension and then the output of lstm is concatenated and passed as input to decoder where it is teacher forced .

**Summary of the model**

****

Works Cited

“dccuchile/spanish-word-embeddings: Spanish word embeddings computed with different methods and from different corpora.” *GitHub*, https://github.com/dccuchile/spanish-word-embeddings. Accessed 2 May 2023.

“GoogleNews-vectors-negative300 ( word2vec ).” *Kaggle*, https://www.kaggle.com/datasets/adarshsng/googlenewsvectors. Accessed 2 May 2023.

“Neural Machine Translation Using seq2seq model with Attention.” *Medium*, 15 June 2021, https://medium.com/geekculture/neural-machine-translation-using-seq2seq-model-with-attention-9faea357d70b. Accessed 2 May 2023.